Open in app ↗

Search

WebSockets

Suresh Podeti

3 min read · Oct 26, 2023

Listen Share More

Photo by Quino Al on Unsplash

In the dynamic world of web applications, responsiveness and real-time communication have become crucial aspects of user experience. Traditional request-response mechanisms, while effective for many scenarios, fall short when it comes to applications that require instant updates and interactive features. This is where WebSockets come into play. In this article, we'll dive into the world of WebSockets, understanding their core concepts such as **handshaking,**

fragmentation, and the ping-pong mechanism, with practical examples to illustrate their importance.

Introduction

WebSockets are a communication protocol that allows **full-duplex, bidirectional communication** between a client (typically a web browser) and a server over a single, **long-lived TCP connection**. Unlike traditional HTTP, which relies on repeated requests and responses, WebSockets enable continuous, real-time data exchange without the overhead of opening and closing connections for each interaction.

Handshaking: The Connection Initialization

At the heart of WebSockets lies the handshaking process. This process establishes a WebSocket connection and transitions it from the standard HTTP protocol to the WebSocket protocol. The client initiates the handshake by sending an HTTP GET request with specific headers, including the "Upgrade" header set to "websocket". The server responds with an HTTP 101 status code (Switching Protocols), indicating its readiness to upgrade the connection. Once the upgrade is complete, the communication switches to the WebSocket protocol.

Fragmentation: Breaking Down Messages

In scenarios where the data payload is large, sending it as a whole in a single frame might not be efficient. This is where fragmentation comes into play. Data can be divided into smaller fragments, each sent as a separate WebSocket frame. The frames in the sequence are marked with the "Fin" (final) bit, which indicates whether the current frame is the last fragment of the message. This mechanism helps in transmitting large payloads without overwhelming the connection.

Ping-Pong Mechanism: Keeping the Connection Alive

To ensure the liveness of the WebSocket connection, a ping-pong mechanism is employed. A ping frame is sent by one party (client or server), and the other party responds with a pong frame. While these frames don't carry any application-specific data, they serve as a heartbeat to check if the connection is still active. If the sender doesn't receive a pong frame within a certain time, it can infer a connection issue.

Examples

1. **Handshaking Example:**
   Imagine a client sends the following HTTP GET request during handshaking:

```
GET /ws-endpoint HTTP/1.1
Host: example.com
Connection: Upgrade
Upgrade: websocket
Sec-WebSocket-Key: <base64-encoded key>
```

2. **Fragmentation Example:**
For a large file transfer, the server might divide the file into smaller chunks, sending each chunk as a separate WebSocket frame. The “Fin” bit indicates whether a frame is the final fragment.

3. **Ping-Pong Example:**
If the client doesn’t receive a pong frame within a certain interval after sending a ping frame, it can take measures like reconnecting or displaying a connection issue message.

Conclusion:
WebSockets have revolutionized real-time communication in web applications by providing a persistent, low-latency connection that enables bidirectional data exchange. Understanding core concepts like handshaking, fragmentation, and the ping-pong mechanism is essential for building responsive and interactive applications that require instant updates. As we continue to embrace the potential of WebSockets, we unlock new possibilities for dynamic and engaging user experiences on the modern web.

Websocket System Design Interview Software Architect Software Architecture

Distributed System Design

Edit profile

Written by Suresh Podeti

1.2K Followers

Aspiring Software Architect | SDE III (Golang) @JungleeGames| Ex-Byju's | Co-founder @Rupant Tech. | Ex-Synopsys | I.I.T Kharagpur | 7+ Years of Experience

Recommended from Medium